AD NUMBER
AD468847
LIMITATION CHANGES
TO: **Approved for public release; distribution is unlimited.**
FROM: **Distribution authorized to U.S. Gov't. agencies and their contractors; Administrative/Operational Use; 26 MAR 1965. Other requests shall be referred to Office of Naval Research, Arlington, VA 22203.**
AUTHORITY
ONR/CNA ltr 13 Sep 1971

SECURITY MARKING

The classified or limited status of this report applies to each page, unless otherwise marked.
Separate page printouts MUST be marked accordingly.

THIS DOCUMENT CONTAINS INFORMATION AFFECTING THE NATIONAL DEFENSE OF THE UNITED STATES WITHIN THE MEANING OF THE ESPIONAGE LAWS, TITLE 18, U.S.C., SECTIONS 793 AND 794. THE TRANSMISSION OR THE REVELATION OF ITS CONTENTS IN ANY MANNER TO AN UNAUTHORIZED PERSON IS PROHIBITED BY LAW.

NOTICE: When government or other drawings, specifications or other data are used for any purpose other than in connection with a definitely related government procurement operation, the U. S. Government thereby incurs no responsibility, nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use or sell any patented invention that may in any way be related thereto.

CNA

COMPLEX TARGET COVERAGE
OEG COMPUTER PROGRAM 13-63P

By R. L. Smith, G. A. Westlund,
P. E. DePoy, R. V. Ridings,
and S. A. Denenberg

Research Contribution No. 68

This research contribution does not necessarily represent the views of CNA or the U.S. Navy. It may be modified or withdrawn at any time.

CONTRACT NONR 3732(00)

Research Contribution

OPERATIONS EVALUATION GROUP

Center for Naval Analyses
THE FRANKLIN INSTITUTE
WASHINGTON 25, D. C.

26 March 1965

RESEARCH CONTRIBUTION

Operations Evaluation Group

CENTER FOR NAVAL ANALYSES

COMPLEX TARGET COVERAGE
OEG COMPUTER PROGRAM 13-63P

By R.L. Smith, G.A. Westlund,
P.E. DePoy, R.V. Ridings,
and S.A. Denenberg

Research Contribution No. 68
(Originally IRM-45)

R. L. Smith
G. A. Westlund
Phil E. DePoy
R. V. Ridings
S. A. Denenberg

ABSTRACT

This research contribution presents a usage manual for an IBM 7090 computer program. The program employs a Monte Carlo simulation to determine the probability of destroying individual point targets within a target complex with one or more groups of weapons. It is assumed that the groups are delivered with a bivariate normal aiming error and that the individual weapons are distributed with an independent bivariate normal ballistic dispersion. The program is designed for conditional damage data for fragmentation generated by an IBM 7090 program furnished by the U.S. Naval Ordnance Test Station (NOTS), China Lake. A flow chart, a listing of the FORTRAN program and a sample problem are included.

TABLE OF CONTENTS

I. INTRODUCTION

Previous OEG computer programs (reference (a)) determine the probability of destroying rectangular or line targets with a single group of weapons delivered with a bivariate normal aiming error, and consider the individual weapons to be distributed about their mean points of impact within the group with a bivariate normal ballistic dispersion. This program determines the probabilities of destroying stationary individual point targets at various locations within a target complex with one or more groups of weapons - making the same assumptions regarding aiming error and ballistic dispersion as were made in the previous programs.

In considering the destruction of point targets, a conditional damage function is required. The conditional damage function is defined in reference (b) as a function which gives the probability that a target which is located at a particular range and bearing from the weapon detonation point will suffer at least the stated degree of damage. Analytical functions which are often used to approximate conditional damage functions are a definite-range (cookie-cutter) function or a Gaussian function (these functions are described in detail in reference (b)). Neither of these are completely adequate however, for describing the effects of fragmentation on a point target. Therefore, this program has been designed to use the output of the IBM 7090 warhead lethality program written at the U.S. Naval Ordnance Test Station (NOTS), China Lake. The NOTS program provides,for each 5 degree sector around the weapon, the average probabilities of destroying a target located within small cells at various radii from the detonation point. It should be noted that the design of the program does not preclude the use of a cookie-cutter or Gaussian conditional damage function if the probabilities are determined for cells at the proper radii (explained in a later section of this research contribution for each 5 degree segment).

II. GENERAL DESCRIPTION

This program is a Monte Carlo simulation for use on the IBM 7090 computer. It determines the probabilities of destroying individual point targets within a target complex on the basis of the following assumptions:

a. The aiming error and the ballistic dispersion are bivariate normal, and are independent in the same coordinates, hereafter called the range and deflection coordinates (the names applied to these coordinates result from a common application of the model to bombing problems in which the range coordinate is in the direction of the flight of the aircraft and the deflection coordinate is normal to the flight path in the horizontal plane).

b. There is no cumulative damage effect, i.e., the probability of destroying the target with an individual weapon is independent of the number and locations of other weapons impacting in the vicinity of the target. For fragmentation effects this assumption is not rigorously valid since the damage criterion is often stated as some minimum number of fragments per unit area with at least a minimum energy impacting on the target. Thus, even if the number of fragments impacting on the target from any one of the weapons might be less than the number specified by the damage criterion, the sum of the fragments from two or more weapons might exceed the minimum number.

c. The conditional damage function is symmetrical about the range coordinate through the weapon.

Limiting inputs to the program are:

a. Number of targets, 100.
b. Number of weapon groups, 10.
c. Number of weapons per group, 100.
d. Number of conditional damage function cells per 5 degree sector, 100. (Corresponding to a maximum radius of effect of 3,920 feet).

The conditional damage data must be provided for cells at various radii from the weapon for each 5 degree sector. The cells - except for the innermost - are made approximately square in shape by selecting the outer radius of each cell to be equal to the inner radius times the factor $(1 + \sin 5^\circ)$. The outer radius of the inner cell for each sector is equal to 1 foot (see figure 1). The value of the conditional damage function is taken as 1 when the radius is less than 1 foot, i.e., for all of the inner cells. The individual cells are denoted by a pair of subscripts, kl, where k refers to the k^{th} sector numbering from the tail of the weapon and l refers to the l^{th} cell in a given sector numbering out from the weapon. The program accepts the NOTS output, in which angles are measured relative to the bomb tail as shown in figure 1. All angles calculated in the program however, are measured from the bomb nose, positive direction counter-clockwise; but because of symmetry about the range coordinate of the bomb, values of the conditional damage function for angles that are measured from the bomb tail can be equated by a simple scheme to angles that are measured from the bomb nose; i.e., a cell having a 10 degree angle measured from the bomb nose would have the same conditional damage value as the corresponding cell in figure 1 having the 175 degree angle.

In addition to the parameters specifying the weapon effects and aiming and ballistic dispersions, the program inputs consist of parameters specifying the target locations, the angle of approach to the target for the delivery of each group of weapons (measured from the X-axis to the R-axis - see figure 2), the reliability of the individual weapons, scale factors to shrink or expand the weapon patterns, the number of dummy passes through the random number generator and the number of Monte Carlo iterations to be made.

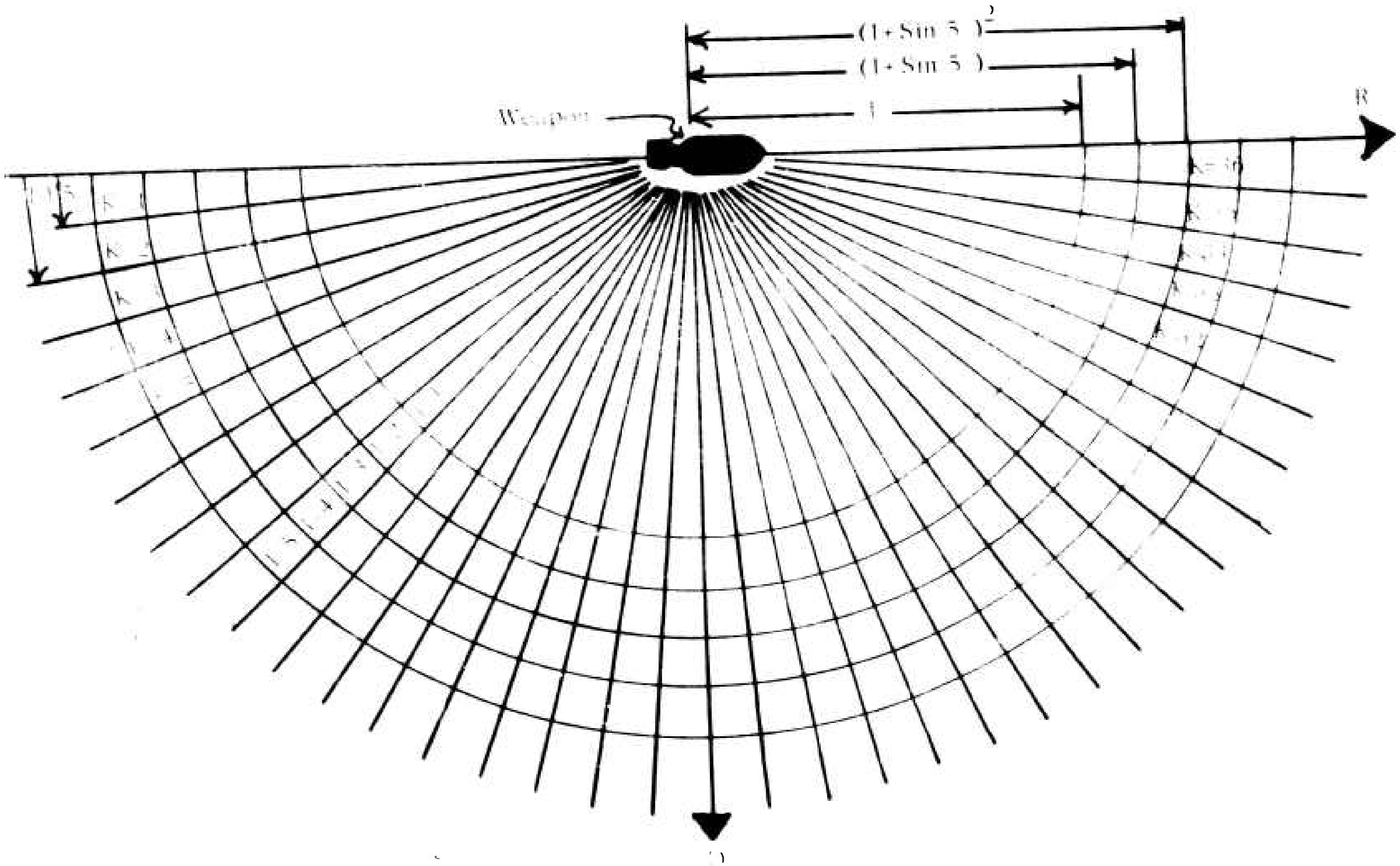
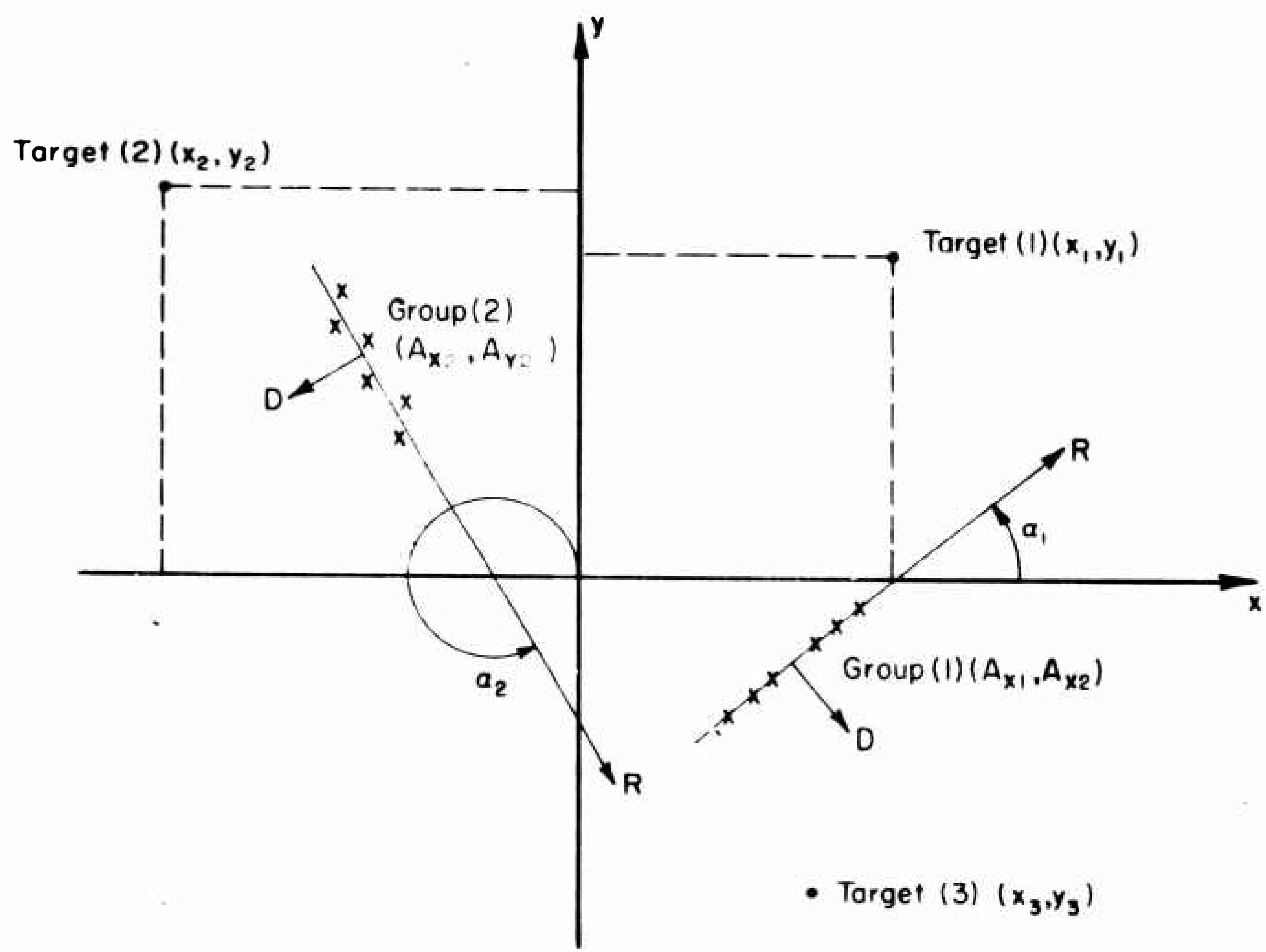


FIGURE 1



y
x
Target (2) (x_2, y_2)
Target (1)(x_1,y_1)
Group(2)
(A_{x2} , A_{y2})
D
R
α_2
α_1
R
Group (1)(A_{x1},A_{x2})
D
Target (3) (x_3,y_3)

FIGURE 2

III. METHOD OF SOLUTION

The inputs for the program consist of the following parameters:

Address*	Symbol	FORTRAN Label	Description
1, 1, 1	N_T	NT	Number of targets
1, 1, 2	N_G	NG	Number of weapon groups
1, 1, 3	I	MCI	Number of Monte Carlo iterations
1, 1, 4	n_r	NEP	Number of empty passes through random number generator
1, 1, 5	N_C	NC	Number of damage probability contours
1, 1, 6	$\mathcal{R}$	REL	Weapon reliability
1, 1, 7	σ_{AR}	SIGR	Range aiming error, standard deviation
1, 1, 8	σ_{AD}	SIGD	Deflection aiming error, standard deviation
1, 1, 9	σ_{BR}	BR	Range ballistic dispersion, standard deviation
1, 1, 10	σ_{BD}	BD	Deflection ballistic dispersion, standard deviation
1, 1, 11	G_R	GR	Range scale factor for weapon group
1, 1, 12	G_D	GD	Deflection scale factor for weapon group
1, 2, 1...1, 2, 10	N_i	N(I)	Number of weapons in i^{th} group
1, 3, 1...1, 3, 10	α_i	ALPHA(I)	Angle of approach for delivery of i^{th} group (degrees), $0 \le \alpha_i < 360$
1, 4, 1...1, 4, 10	A_{Xi}	AX(I)	Aiming point of i^{th} group X-coordinate
1, 5, 1...1, 5, 10	A_{Yi}	AY(I)	Aiming point of i^{th} group Y-coordinate
1, 6, 1...1, 6, 100	X_m	X(M)	Target location of m^{th} target, X-coordinate
1, 7, 1...1, 7, 100	Y_m	Y(M)	Target location of m^{th} target, Y-coordinate
2, 1, 1...2, 1, 100 ⋮ 2, 10, 1...2, 10, 100	Δ_{Rij}	DELR(I, J)	Aimpoint range displacement of j^{th} weapon of i^{th} group from group aiming point

*The parameter addresses are explained in section IV, User's Instructions.

Address	Symbol	FORTRAN Label	Description
3, 1, 1...3, 1, 100	Δ_{Dij}	DELD(I, J)	Aimpoint deflection displacement of j^{th} weapon of i^{th} group from group aiming point
⋮			
3, 10, 1...3, 10, 100			
4, 1, 1...4, 1, 100	C_{kl}	C(K, L)	Probability that a target located in the k^{th} sector of the l^{th} annulus is destroyed
⋮			
4, 36, 1...4, 36, 100			

Values for addresses 4, 1, 1 through 4, 36, 1 need not be entered by the user since the program itself sets these values to 1.0. The user may, however, enter whatever values he wishes for these cells and they will be used by the program.

The program flow chart and FORTRAN statements are included as appendixes A and B. The solution is obtained in the following manner:

1. Before starting the first iteration for the first data set, n_r dummy passes are made through the random number generator (see appendix C). Thus, if the same data are run more than one time, a new set of random numbers can be selected for each run.

2. For each weapon in each group, the displacement from the group reference point (Δ_R, Δ_D) is adjusted with the scale factors (G_R, G_D) to spread or shrink the pattern:

$$\delta_{R_{ij}} = G_R \, \Delta_{R_{ij}} \tag{1}$$

$$\delta_{D_{ij}} = G_D \, \Delta_{D_{ij}} \tag{2}$$

3. For each Monte Carlo iteration:

A. Given the group aimpoint (A_{Xi}, A_{Yi}), the standard deviations of aim error (σ_{AR}, σ_{AD}), two random numbers (n_1, n_2) selected from a standard normal distribution (zero mean and unit variance) and the approach angle α_i, the coordinates for the pattern impact reference point relative to the X, Y axes are determined for each group:

$$R_{Xi} = A_{Xi} + n_1 \, \sigma_{AR} \cos \alpha_i + n_2 \, \sigma_{AD} \sin \alpha_i \tag{3}$$

$$\text{and for } 0 \le R < 1, \quad l = 1$$

$$R \ge 1, \quad l = \left[\frac{\ln R}{\ln(1 + \sin 5^\circ)} + 2\right] \tag{10}$$

(where $[x]$ is defined as the largest integer less than or equal to x).

F. From the conditional damage data for the kl^{th} cell, C_{kl} gives the probability that the m^{th} target is destroyed by the ij^{th} weapon. Given the reliability ($\mathcal{R}$), the probability that the m^{th} target is destroyed by at least one of the weapons is:

$$t_m = 1 - \prod_{i=1}^{N_G} \prod_{j=1}^{N_i} (1 - \mathcal{R}\, C_{kl}) \tag{11}$$

4. The Monte Carlo iteration is repeated (i) times and the values of t_m are accumulated for each target. At the conclusion, the estimate of the probability of destroying each target is determined by:

$$p_m = \frac{\sum_{N=1}^{I} t_m}{I} \tag{12}$$

IV. USER'S INSTRUCTIONS

An OEG Subroutine, 1-63S, called DATA (see appendix E) is used to read the punched card data sets into the computer. A feature of this subroutine is that data are identified and stored in the computer memory through the use of addresses that are the subscripts of the internal data array.

The addresses shown in section III are punched into cards together with the values of the parameters stored at those addresses. A data form which contains the addresses and parameter values is shown in section V for the sample problem. Any number of successive data sets may be submitted at one time as long as the data sets are separated by one blank card and the last data set in the pack is followed by two blank cards. Successive data sets may contain cards for only those parameters that have values different from the preceding data set.

V. SAMPLE PROBLEM

Four targets located at (1, 2), (3, -3), (-1, 1) and (-2, -1) are to be attacked with two groups of weapons of two each. Each weapon has a "cookie-cutter" damage function with a radius of effect of 1. The standard deviations of aim error are 3 in the range coordinate and 1 in deflection. The ballistic dispersion

$$R_{Yi} = A_{Yi} + n_1 \sigma_{AR} \sin \alpha_i - n_2 \sigma_{AD} \cos \alpha_i \qquad (4)$$

$$(i = 1, 2, \ldots, N_G)$$

B. For each weapon in the group, the group impact reference point (R_{Xi}, R_{Yi}), the weapon displacement from the reference point (δ_{Rij}, δ_{Dij}), the standard deviations of ballistic dispersion (σ_{BR}, σ_{BD}), two standard normal random numbers (n_3, n_4) for each weapon, and the approach angle α_i, the coordinates of the weapon impact point relative to the X, Y axes are determined:

$$X_{ij} = R_{Xi} + (n_1 \sigma_{BR} + \delta_{Rij}) \cos \alpha_i + (n_2 \sigma_{BD} + \delta_{Dij}) \sin \alpha_i \qquad (5)$$

$$Y_{ij} = R_{Yi} + (n_1 \sigma_{BR} + \delta_{Rij}) \sin \alpha_i - (n_2 \sigma_{BD} + \delta_{Dij}) \cos \alpha_i \qquad (6)$$

$$(j = 1, 2, \ldots, N_i)$$

C. For each target, the separation distance (squared) from each weapon is determined using the target location (X_m, Y_m) and the weapon impact points (X_{ij}, Y_{ij}):

$$R^2 = (X_{ij} - X_m)^2 + (Y_{ij} - Y_m)^2 \qquad (7)$$

D. If the square of separation distance, R^2, is less than the square of maximum radius of effect of the weapon, R^2_{max}, the relative bearing of the target from the weapon is determined using an OEG Subroutine, ACOSD (see appendix D).

$$\theta = \left| \cos^{-1} \left(\frac{X_m - X_{ij}}{R} \right) - \alpha_i \left(\frac{Y_m - Y_{ij}}{|Y_m - Y_{ij}|} \right) \right| \qquad (8)$$

E. The conditional damage contour cell numbers, k and l, are then determined as follows:

$$\begin{array}{lll} \text{for } 0^\circ < \theta \le 5^\circ, & k = 1 & \\ 5^\circ < \theta \le 10^\circ, & k = 2 & \qquad (9) \\ \vdots & \vdots & \\ 175^\circ < \theta \le 180^\circ, & k = 36 & \end{array}$$

is 1 in range and 0.5 in deflection. The reliability of the weapons is 0.8. The centers of both groups are aimed at the origin of the x-y axis (0,0) and the weapons are spaced along the range coordinate 2 units apart. One group is delivered at an angle of 30° to the x-axis, the other at an angle of 150° to the x-axis.

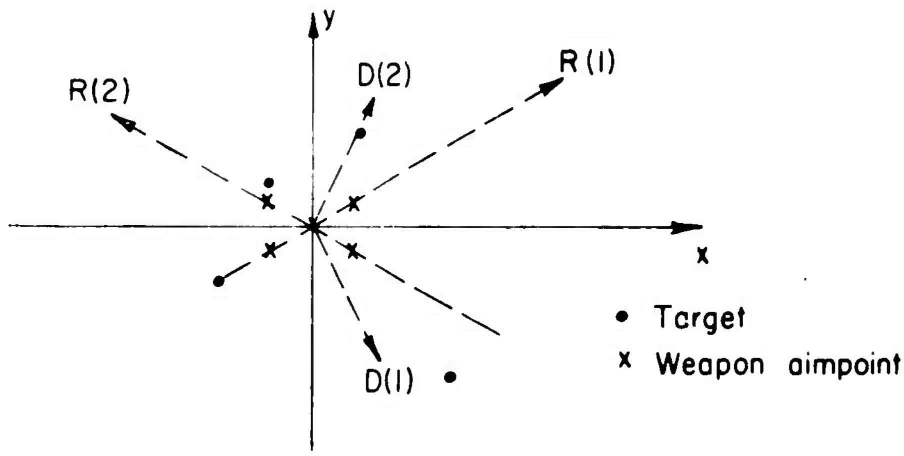


FIG. 3: SAMPLE PROBLEM

The input data are shown on the attached sample data form.

VI. KEYPUNCH INSTRUCTIONS

Keypunch instructions for the use of DATA Subroutine, 1-63S, are contained in appendix E.

VII. OPERATOR'S INSTRUCTIONS

Run under control of the Bell System on the IBM 7090. No special instructions.

VIII. TIMING

This program requires approximately:

$$\frac{1}{150} \times N_T \times I \times \sum_{i=1}^{N_G} N_i \text{ seconds}$$

where

N_T = number of targets

I = number of Monte Carlo iterations

$\sum_{i=1}^{N_G} N_i$ = number of weapons in all groups summed.

CLUSTER TARGET INPUTS

CLUSTERS	TARGETS	WEAPON GROUPS	ITERATIONS	RELIABILITY	RANDOM NUMBER PASSES
1	4	2	3000	0.8000	114

	RANGE	DEFLECTION
AIMING ERROR, FEET	3.0	1.0
BALLISTIC ERROR, FEET	1.0	0.5
GROUP SCALE FACTOR	1.0	1.0

GROUP NUMBER	AIMING POINTS X	AIMING POINTS Y	APPROACH ANGLE, DEGREES	NUMBER OF WEAPONS
1	0.	0.	30.0	2
2	0.	0.	150.0	2

GROUP NUMBER	WEAPON NUMBER	AIMPOINT DISPLACEMENTS, FEET RANGE	DEFLECTION
1	1	1.0	0.
1	2	-1.0	0.
2	1	1.0	0.
2	2	-1.0	0.

DESTRUCTION PROBABILITY CONTOURS

K	1
1	1.00000
2	1.00000
3	1.00000
4	1.00000
5	1.00000
6	1.00000
7	1.00000
8	1.00000
9	1.00000
10	1.00000
11	1.00000
12	1.00000
13	1.00000
14	1.00000
15	1.00000
16	1.00000
17	1.00000
18	1.00000
19	1.00000
20	1.00000
21	1.00000
22	1.00000
23	1.00000
24	1.00000
25	1.00000
26	1.00000
27	1.00000
28	1.00000
29	1.00000
30	1.00000
31	1.00000
32	1.00000
33	1.00000
34	1.00000
35	1.00000
36	1.00000

TARGET NUMBER	LOCATIONS X	Y	KILL PROBABILITY
1	1.0	2.0	0.118
2	3.0	-3.0	0.059
3	-1.0	1.0	0.239
4	-2.0	-1.0	0.184

CNA COMPUTER DATA SUBMITTAL FORM

Submitted by: J. Doe Date: 1 August 1964

Program No. 13-63 P Est. Time 4 min. Classification Uncl.

Special Instructions:

Address	Value	Address	Value	Address	Value	Address	Value
1,1,1	4	1,6,1	1				
1,1,2	2	1,7,1	2				
1,1,3	3000	1,6,2	3				
1,1,4	114	1,7,2	-3				
1,1,5	1	1,6,3	-1				
1,1,6	.8	1,7,3	1				
1,1,7	3	1,6,4	-2				
1,1,8	1	1,7,4	-1				
1,1,9	1	2,1,1	1				
1,1,10	.5	3,1,1	0				
1,1,11	1	2,1,2	-1				
1,1,12	1	3,1,2	0				
1,2,1	2	2,2,1	1				
1,3,1	30	3,2,1	0				
1,4,1	0	2,2,2	-1				
1,5,1	0	3,2,2	0				
1,2,2	2	—	b —				
1,3,2	150	—	b —				
1,4,2	0						
1,5,2	0						

NOTES:

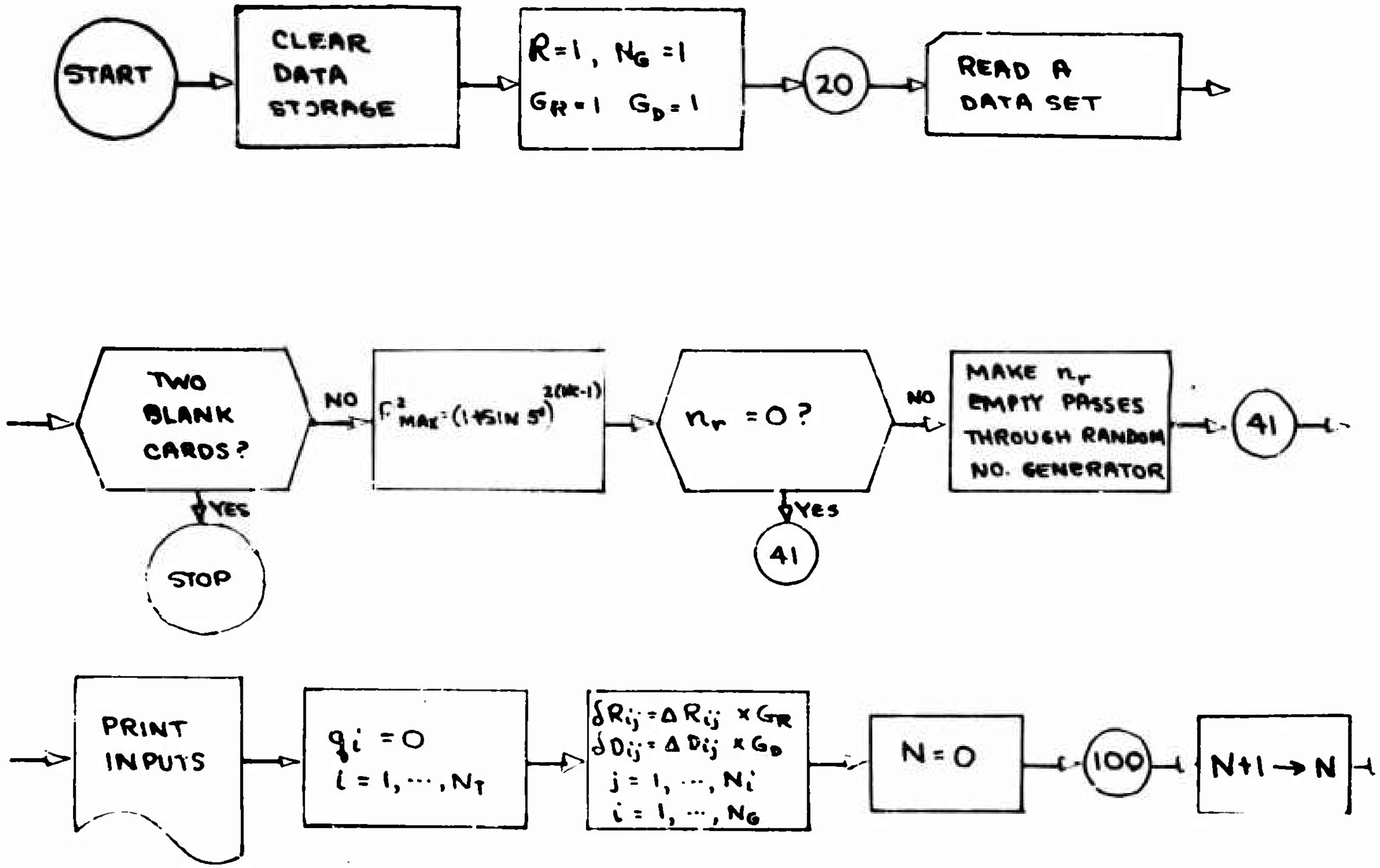
1. A value of zero must be entered as 0, not left blank.
2. Decimal pts. may be omitted if understood to follow the rightmost digit.
3. The value 3×10^{-5} may be entered as .00003 or 3-5, not as 3×10^{-5}.
4. The factor portion of a value may not contain more than 8 digits.
5. The exponent portion of a value must lie within the range ±39.
6. Exponents may be omitted if zero. If not, they must be signed.
7. Blank cards should be indicated by: ——————b—————— .

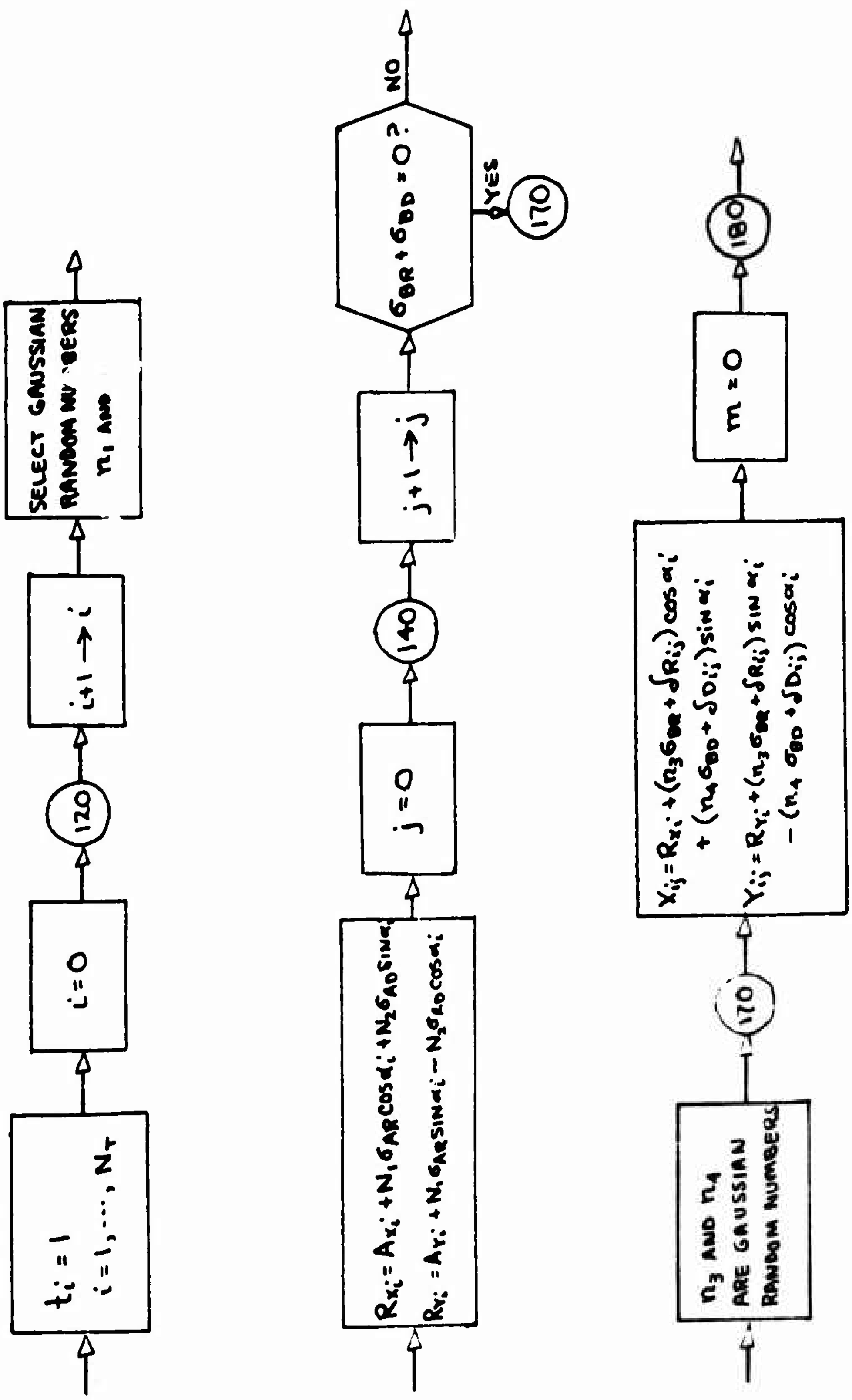
References: (a) NAVWAG Interim Research Memorandum, NIRM-12; "Usage Manual for a Computer Program to Compute the Effectiveness of Groups of Weapons against Rectangular and Line Targets" Unclassified 11 Dec 1959

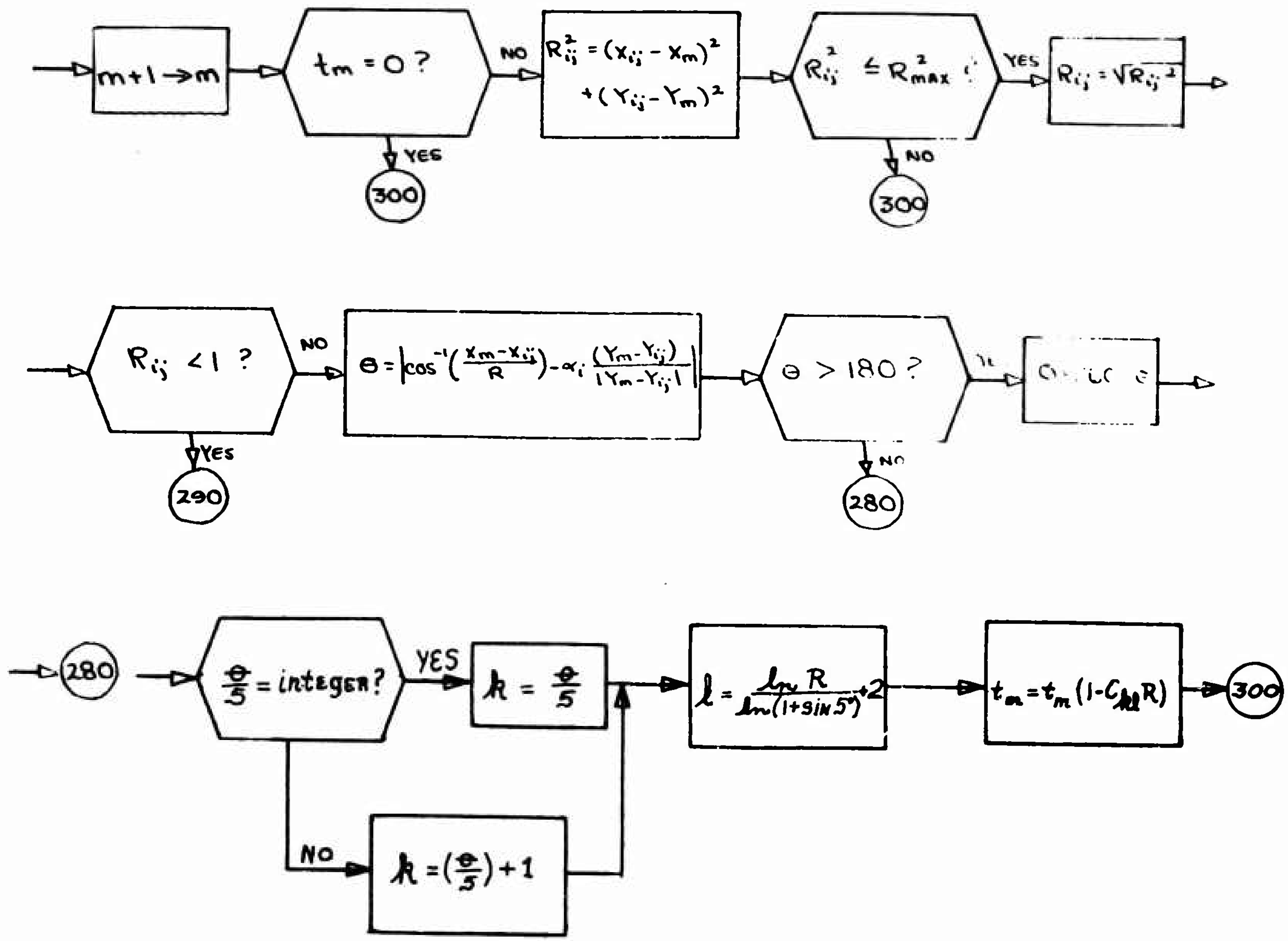
(b) OEG Study 626, "Probability-of-Damage Problems of Frequent Occurrence" Unclassified 11 Dec 1959

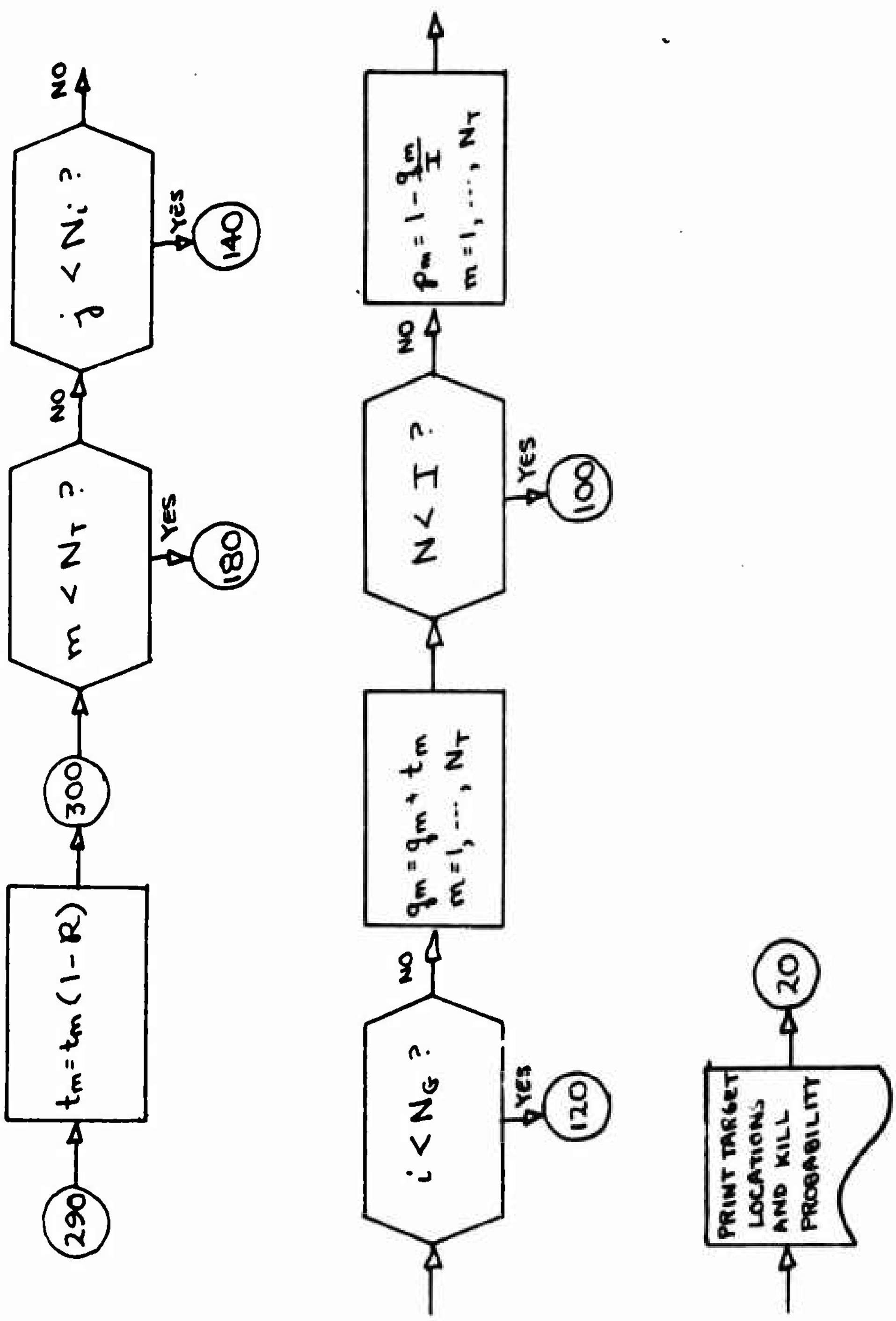
APPENDIX A

FLOW CHART


START
CLEAR DATA STORAGE
R=1, N_G=1
G_R=1 G_D=1
20
READ A DATA SET
TWO BLANK CARDS?
YES
STOP
NO
Γ²MAX=(1+SIN S°)^2(K-1)
n_r =0?
YES
41
NO
MAKE n_r EMPTY PASSES THROUGH RANDOM NO. GENERATOR
41
PRINT INPUTS
q_i =0
i = 1, ···, N_T
δR_ij=ΔR_ij ×G_R
δD_ij=ΔD_ij ×G_D
j = 1, ···, N_i
i = 1, ···, N_G
N=0
100
N+1→N


ti = 1
i = 1, ..., NT
i = 0
120
i+1 → i
SELECT GAUSSIAN
RANDOM NUMBERS
n1 AND
RXi = AXi + N1 σAR cos αi + N2 σAD sin αi
RYi = AYi + N1 σAR sin αi − N2 σAD cos αi
j = 0
140
j+1 → j
σBR + σBD = 0?
YES
170
NO
n3 AND n4
ARE GAUSSIAN
RANDOM NUMBERS
170
Xij = RXi + (n3 σBR + δRij) cos αi
+ (n4 σBD + δDij) sin αi
Yij = RYi + (n3 σBR + δRij) sin αi
− (n4 σBD + δDij) cos αi
m = 0
180


m+1 → m
t_m = 0 ?
YES
300
NO
R_ij² = (X_ij − X_m)² + (Y_ij − Y_m)²
R_ij² ≤ R_max² ?
NO
300
YES
R_ij = √R_ij²
R_ij < 1 ?
YES
290
NO
θ = |cos⁻¹((X_m − X_ij)/R) − α_i (Y_m − Y_ij)/|Y_m − Y_ij||
θ > 180 ?
NO
280
280
Φ/5 = integer?
YES
k = Φ/5
NO
k = (Φ/5) + 1
300


290
$t_m = t_m(1-R)$
300
$m < N_T$?
YES
180
NO
$j < N_i$?
YES
140
NO
$i < N_G$?
YES
120
NO
$q_m = q_m + t_m$
$m = 1, \ldots, N_T$
$N < I$?
YES
100
NO
$p_m = 1 - \frac{q_m}{I}$
$m = 1, \ldots, N_T$
PRINT TARGET LOCATIONS AND KILL PROBABILITY
20

APPENDIX B

PROGRAM LISTING

```
      DIMENSION D(4,36,100), DELR(10,100), DELD(10,100), P(100), Q(100),
     1T(100), X(100), Y(100), N(10), ALPHA(10), AX(10), AY(10), COSA(10)
     2, SINA(10), RN(2),C(4,36,100),LL(10)
      EQUIVALENCE (D,C),(D(721),REL),(D(865),SIGR),(D(1009),SIGD),
     1(D(1153),BR),(D(1297),BD),(D(1441),GR),(D(1585),GD)
      DO 10 I=1,14400
   10 D(I)=0.0
      DO 15 K=1,36
   15 C(4,K,1)=1.0
      REL=1.0
      GR=1.0
      GD=1.0
      NG=1
   20 CALL DATA (C,4,36,100,IND)
      IF (IND-1) 30,420,20
   30 NT=D(1,1,1)
      NG=D(1,1,2)
      MCI=D(1,1,3)
      NEP=D(1,1,4)
      NC=D(1,1,5)
      ATERM=1.0+SINDF(5.0)
      NCX=2*(NC-1)
      RMAXSQ=ATERM**NCX
      IF (NEP) 41,41,35
   35 DO 40 I=1,NEP
   40 CALL GRNUMB (DUMMY)
      D(1,1,4)=0.0
   41 DO 45 I=1,NG
      N(I)=D(1,2,I)
      ALPHA(I)=D(1,3,I)
      AX(I)=D(1,4,I)
      AY(I)=D(1,5,I)
      II=N(I)
      DO 45 J=1,II
      DELR(I,J)=D(2,I,J)
   45 DELD(I,J)=D(3,I,J)
      PRINT 50
   50 FORMAT (1H1, 38X, 41HC O M P L E X   T A R G E T   I N P U T S)
      PRINT 51
   51 FORMAT (////13X, 8HCONTOURS, 5X, 7HTARGETS, 5X, 13HWEAPON GROUPS,
     1 5X, 10HITERATIONS, 5X, 11HRELIABILITY, 5X, 20HRANDOM NUMBER PASS
     2ES)
      PRINT 52, NC, NT, NG, MCI, REL, NEP
   52 FORMAT (//15X, I4, 9X, I3, 12X, I2, 13X, I5, 10X, F6.4, 15X, I5)
      PRINT 53
   53 FORMAT (////42X,5HRANGE,15X,10HDEFLECTION)
      PRINT 54, SIGR, SIGD
   54 FORMAT (//10X, 18HAIMING ERROR, FEET,13X, F6.1, 17X, F6.1)
      PRINT 55, BR, BD
   55 FORMAT (10X, 21HBALLISTIC ERROR, FEET,10X, F6.1, 17X, F6.1)
      PRINT 56, GR, GD
   56 FORMAT (10X, 18HGROUP SCALE FACTOR,13X, F6.1, 17X, F6.1)
      PRINT 57
   57 FORMAT (//////12X, 5HGROUP,14X, 13HAIMING POINTS,11X, 15HAPPROACH
     1ANGLE,,9X, 9HNUMBER OF)
      PRINT 58
   58 FORMAT (12X, 6HNUMBER, 11X, 1HX, 15X, 1HY, 13X, 7HDEGREES, 14X,
     17HWEAPONS)
      PRINT 59
```

```
 59 FORMAT (1H0)
    DO 60 I=1,NG
 60 PRINT 61, I, AX(I), AY(I), ALPHA(I), N(I)
 61 FORMAT (13X, I2, 11X, F6.1, 10X, F6.1, 12X, F5.1, 16X, I3)
    PRINT 62
 62 FORMAT (1H1, 12X, 5HGROUP, 7X, 6HWEAPON, 6X, 28HAIMPOINT DISPLACEM
   1ENTS, FEET)
    PRINT 63
 63 FORMAT (13X,6HNUMBER,6X,6HNUMBER,9X,5HRANGE,7X,10HDEFLECTION)
    PRINT 59
    DO 64 I=1,NG
    PRINT 59
    II=N(I)
    DO 64 J=1,II
 64 PRINT 65,I, J, DELR(I,J), DELD(I,J)
 65 FORMAT (14X, I2, 9X, I3, 9X, F6.1, 9X, F6.1)
    NCP=NC
    JL=-9
    JU=0
 67 IF (NCP-10) 68,68,69
 68 JU=JU+NCP
    NCP=0
    GO TO 70
 69 JU=JU+10
    NCP=NCP-10
 70 JL=JL+10
    PRINT 71
 71 FORMAT (1H1, 41X, 32HDESTRUCTION PROBABILITY CONTOURS)
    LL(1)=JL
    DO 72 I=2,10
 72 LL(I)=LL(I-1)+1
    LLL=JU-JL+1
    PRINT 73, (LL(I),I=1,LLL)
 73 FORMAT (1H0, 10X, 1HK, 6X, 10(I3,7X))
    DO 74 KK=1,36
 74 PRINT 75, KK, (C(4,KK,LNC), LNC=JL,JU)
 75 FORMAT (10X, I2, 6X, 10(F7.5, 3X))
    IF (NCP) 76,76,67
 76 NIT=0
    DO 80 I=1,NT
 80 Q(I)=0.0
    DO 90 I=1,NG
    COSA(I)=COSDF(ALPHA(I))
    SINA(I)=SINDF(ALPHA(I))
    JJ=N(I)
    DO 90 J=1,JJ
    DELR(I,J)=DELR(I,J)*GR
 90 DELD(I,J)=DELD(I,J)*GD
    DO 95 M=1,NT
    X(M)=D(1,6,M)
 95 Y(M)=D(1,7,M)
100 NIT=NIT+1
    DO 110 M=1,NT
110 T(M)=1.0
    IG=0
120 IG=IG+1
    DO 130 I=1,2
130 CALL GRNUMB(RN(I))
```

```
      RX=AX(IG)+RN(1)*SIGR*COSA(IG)+RN(2)*SIGD*SINA(IG)
      RY=AY(IG)+RN(1)*SIGR*SINA(IG)-RN(2)*SIGD*COSA(IG)
      JW=0
  140 JW=JW+1
      IF (BR+BD) 170,170,150
  150 DO 160 I=1,2
  160 CALL GRNUMB(RN(I))
  170 XX=RX+(RN(1)*BR+DELR(IG,JW))*COSA(IG)+(RN(2)*BD+DELD(IG,JW))*
     1SINA(IG)
      YY=RY+(RN(1)*BR+DELR(IG,JW))*SINA(IG)-(RN(2)*BD+DELD(IG,JW))*COSA
     1(IG)
      MT=0
  180 MT=MT+1
      IF (T(MT)) 300,300,190
  190 RSQ=(XX-X(MT))**2+(YY-Y(MT))**2
      IF (RSQ-RMAXSQ) 200,200,300
  200 R=SQRTF(RSQ)
      IF (R-1.0) 290,210,210
  210 THETA=ABSF(ACOSDF((X(MT)-XX)/R)-SIGNF(ALPHA(IG),Y(MT)-YY))
  260 IF (THETA-180.0) 280,280,270
  270 THETA=360.0-THETA
  280 IF(MODF(THETA,5.)) 282,281,282
  281 K=THETA/5.
      GO TO 283
  282 K=THETA/5.+1.
  283 L=LOGF(R)/LOGF(ATERM)+2.0
      K37=37-K
      T(MT)=T(MT)*(1.0-C(4,K37,L)*REL)
      GO TO 300
  290 T(MT)=T(MT)*(1.0-REL)
  300 IF (MT-NT) 180,310,310
  310 IF (JW-N(IG)) 140,320,320
  320 IF (IG-NG) 120,330,330
  330 DO 340 M=1,NT
  340 Q(M)=Q(M)+T(M)
      IF (NIT-MCI) 100,350,350
  350 DO 360 M=1,NT
  360 P(M)=1.0-Q(M)/FLOATF(MCI)
      PRINT 370
  370 FORMAT  (1H1, 12X, 6HTARGET, 10X, 9HLOCATIONS, 14X, 4HKILL)
      PRINT 380
  380 FORMAT (13X, 6HNUMBER, 7X, 1HX, 13X, 1HY, 10X, 11HPROBABILITY)
      PRINT 59
      K=0
      DO 410 M=1,NT
      K=K+1
      PRINT 390, M, X(M), Y(M), P(M)
  390 FORMAT (13X, I3, 6X, F6.1, 8X, F6.1, 7X, F7.3)
      IF (K-10) 410,400,400
  400 PRINT 59
      K=0
  410 CONTINUE
      GO TO 20
  420 PRINT 430
  430 FORMAT (1H1)
      CALL ENDJOB
      END
```

APPENDIX C

GRNUMB SUBROUTINE

1. Purpose:

GRNUMB provides a floating point pseudo-random number X The distribution of successive values of X are Gaussian with a mean of zero and a standard deviation of one.

2 Method:

Consider the set of uniformly distributed pseudo-random numbers Y_i . GRNUMB generates a sequence of Y_i by the method of congruences:

$$Y_i = 2^{-35} (5^{15} 2^{35} Y_{i-1} \text{, mod } 2^{35})$$

over the range $0 \le Y_i < 1$. The variance of this uniform set is

$$\sigma_Y{}^2 = \int_0^1 (Y-1/2)^2 \, dY = 1/12 \; .$$

If X is the mean of any selection of m of the uniform numbers Y , the Central Limit Theorem states that the variable X approaches a normal distribution where m is sufficiently large. A satisfactory value for m is 30. Values of X are generated as a sequence of X_n , where n denotes the n^{th} entry to GRNUMB.

$$X_n = \sqrt{1/m\sigma_Y{}^2} \sum_{i=1}^{m} (Y_i - 1/2) = \sqrt{.4} \sum_{i=1}^{30} (Y_i - 1/2)$$

where $Y_o = X_{n-1}$, and $X_o = 2^{-35}$. The variance of this normal set is 1 .

3. Usage:

X is obtained by use of the statement:

CALL GRNUMB (X)

in a FORTRAN program for the IBM 7090.

4. Coding Information:

See the symbolic listing on the following page. GRNUMB is written in the 7090 FAP language It requires 40 words storage space and 900 microseconds operating time.

SYMBOLIC LISTING

```
            FAP
            REM GRNUMB G. WESTLUND    18 JUNE 1962  (7090)
            REM GAUSSIAN DISTRIBUTED RANDOM NUMBER GENERATOR.
            REM ENTER VIA FORTRAN STATEMENT      CALL GRNUMB (X)
            REM SEQUENCE STARTS AT DEC 1, YIELDS X WITH STAND. DEV. =1.
            ENTRY GRNUMB
GRNUMB      SXA XX1, 1
            CLA 1, 4
            STA F
            AXT 30, 1
            STZ NUM
C           LDQ NUMB
            MPY MULT
            STQ NUMB
            CLA NUMB
            SUB CHAR
            ARS 4
            ADD NUM
            STO NUM
            TIX C, 1, 1
            LDQ NUM
            MPY MAGIC
            LRS 27
            TZE D
            LRS 8
            CLA H125
            ADD H8
            LLS 8
            ALS 19
            TRA E
D           CLA H125
            ALS 27
E           STO NUM
            CLA H125
            LLS 27
            FAD NUM
F           STO **
XX1         AXT **, 1
            TRA 2, 4
NUM         HTR **
NUMB        DEC 1
MULT        DEC 30517578125
CHAR        TIX 0, 0, 0
MAGIC       DEC 0.31622780B0
H8          DEC 8
H125        DEC 125
            END
```

APPENDIX D

ASIN SUBROUTINE

1. Purpose:

Given the floating point argument X, ASIN provides a floating point number Y, where

$Y = \sin^{-1} X$	or $Y = \cos^{-1} X$
$-1 \le X \le 1$	$-1 \le X \le 1$
$-\pi/2 \le Y \le \pi/2$	$0 \le Y \le \pi$

Depending upon the entry to the subroutine, Y may be found in radians or degrees.

2. Method:

A series approximation (Hastings), is used to find the Arc cosine. If the Arc sine is desired, the arc cosine is subtracted from $\pi/2$. The result is then converted from radians to degrees if desired.

$$\text{Arcsin } X = \frac{\pi}{2} - \sqrt{1-X}\ \psi(X)$$

$$\psi(X) = a_0 + a_1 X + a_2 X^2 + \ldots + a_7 X^7$$

a_0 = 1.5707, 9630, 50	a_4 = .0308, 9188, 10
a_1 = .2145, 9880, 16	a_5 = -.0170, 8812, 56
a_2 = .0889, 7898, 74	a_6 = .0066, 7009, 01
a_3 = -.0501, 7430, 46	a_7 = -.0012, 6249, 11

degrees = 57.2957, 7951, 3 radians

3. Usage:

Y is obtained by use of one of the four expressions below in a FORTRAN program for the IBM 7090.

ASINF(X)	Arc sine X (radians)
ACOSF(X)	Arc cosine X (radians)
ASINDF(X)	Arc sine X (degrees)
ACOSDF(X)	Arc cosine X (degrees)

4. Coding Information:

See the symbolic listing on page D-2. ASIN is written in the 7090 FAP language. It requires 51 words storage space and 850 microseconds operating time.

SYMBOLIC LISTING

```
        FAP
        REM 3-62S ASIN   G. WESTLUND   7090
        REM ARCSINE-ARCCOSINE SUBROUTINE.  ENTRY VIA FORTRAN
        REM EXPRESSION ASIN(X) OR ACOS(X) YIELDS RADIANS.
        REM ASIND(X) OR ACOSD(X) YIELDS DEGREES.
        REM ARCSINE RANGE = -90 to +90, ARCCOSINE RANGE = 0 TO 180.
        ENTRY ASIN, ACOS, ASIND, ACOSD
ASIND   SXD S, 4
ACOSD   SXA S, 4
        TRA *+2
ASIN    SXD S, 4
ACOS    SXA X4, 4
        STO SIGN
        SSP
        TZE D
        STO X
        CLA ONE
        FSB X
        TZE C
A       CALL SQRT
        STO R
        AXT 7, 4
        LDQ T+7
B       FMP X
        FAD T+7, 4
        XCA
        TIX B, 4, 1
        FMP R
C       FAD PI2
        LDQ SIGN
        LLS 0
D       LXD S, 4
        TXH E, 4, 0
        CHS
        FAD PI2
E       LXA S, 4
        TXL X4, 4, 0
        XCA
        FMP CONV
X4      AXT **, 4
        STZ S
        TRA 1, 4
S       HTR O
X       HTR **
```

SYMBOLIC LISTING (Cont'd)

```
SIGN    HTR **
ONE     DEC 1.
PI2     DEC 1.5707963
R       HTR **
T       DEC -.0066700901, .0170881256, -.0308918810, .0501743046
        DEC -.0889789874, .2145988016, -1.570796305, .0012624911
CONV    DEC 57.295779513
        END
```

APPENDIX E

DATA SUBROUTINE

1. Introduction:

Many computer programs require the flexibility of varying any or all of the parameters in a computer run. Although FORTRAN is fairly flexible in its arithmetic and control statements, its input-output statements are quite rigid. In order to read cards for instance, considerable effort must be expended by the FORTRAN programmer in writing his input statements. This subroutine eliminates some of that tedium. The concept of a "data set" is used. A data set consists of a sequence of punched cards terminated by one blank card. A parameter deck for a computer run may consist of several data sets. Such a parameter deck is terminated by two blank cards.

2. Parameter Addresses:

The primary advantage of this subroutine over FORTRAN input statements results from the use of "parameter addresses." An address is a relative location in the computer memory. It is the subscript of an array or **matrix.** For example, in an array called X, the parameter value X_{53} would be located at address 53. By using the parameter addresses, a user of the program need submit only those parameter values in a data set that are different from those in the previous set.

Three types of addresses are permitted by this subroutine.

(1) A numeric address consisting of one to five characters, each of which is a digit 0 - 9. Such an address (n) refers to the n^{th} element in a specified array.

(2) An alpha address consisting of one to six characters, the first of which must be alphabetic (A-Z). The remaining may be alphabetic or numeric (A-Z or 0-9). Such an address refers to the n^{th} element in a specified array ($1 \leq n \leq 26$), where the first character of the address corresponds to n as the 26 letters of the alphabet correspond to the integers 1-26.

(3) A matrix address consisting of two or more numeric fields separated by commas. For example, the address 53, 47 refers to the element in the 53^{rd} row and the 47^{th} column of a two-dimensional matrix. There is no limit to the number of dimensions in a matrix address.

3. Input Card Format:

A standard submittal form (see attachment) has been designed for the analyst. This form provides for entering parameter values with their associated addresses. The user indicates blank cards to separate data sets. The keypunch operator has the option of punching one address and value per card, or, if the addresses are sequential, of punching one address and several values on a card.

Only columns 1-72 of a card are used. Each column must contain one of the following: a digit (0-9), a "+" or " - " sign or a dash, a letter (A-Z), a period, a comma, or a blank. Each punched card must contain one parameter address. The address may start in column 1, or, if desired, may start in a later column, provided all columns before it are blank. The address is terminated by at least one blank column. Only one address is permitted on the card. Succeeding columns contain one or more parameter values, each separated by one or more blank columns. A value may be signed or unsigned. The length of the value field is variable. No blanks are permitted within a value field. A value may be punched with or without an exponent. An exponent is recognized by the presence of a plus or minus sign (or dash) between the fractional part and exponent part of the value. Decimal points (periods) may be punched in either the fractional or exponent parts of a value. If more than one value is punched on a card, those after the first will be entered at sequential addresses relative to the address of the first value.

4. Usage:

A data set is read by the use of the statement:

CALL DATA (X, I)

in a FORTRAN program for the IBM 7090. The argument X is the name of an array in the program. The argument I is an indicator set by the subroutine. This indicator may be tested by the main program upon return from the subroutine. It will have a value of 0 or 1 or 2.

0: The subroutine has read a data set. The main program will normally proceed to operate on this data.

1: The subroutine has read the second blank card which terminates the parameter deck. The main program will normally terminate at this point.

2: The subroutine has read a "bad" data card. The main program may terminate the run, or ignore the card and return to the subroutine to read the rest of the data set.

If the cards to be read contain matrix addresses, additional arguments must be included in the FORTRAN calling statement:

CALL DATA (X, D_1, D_2, D_3, ..., D_n, I)

where D_i is the i^{th} dimension of the matrix X.

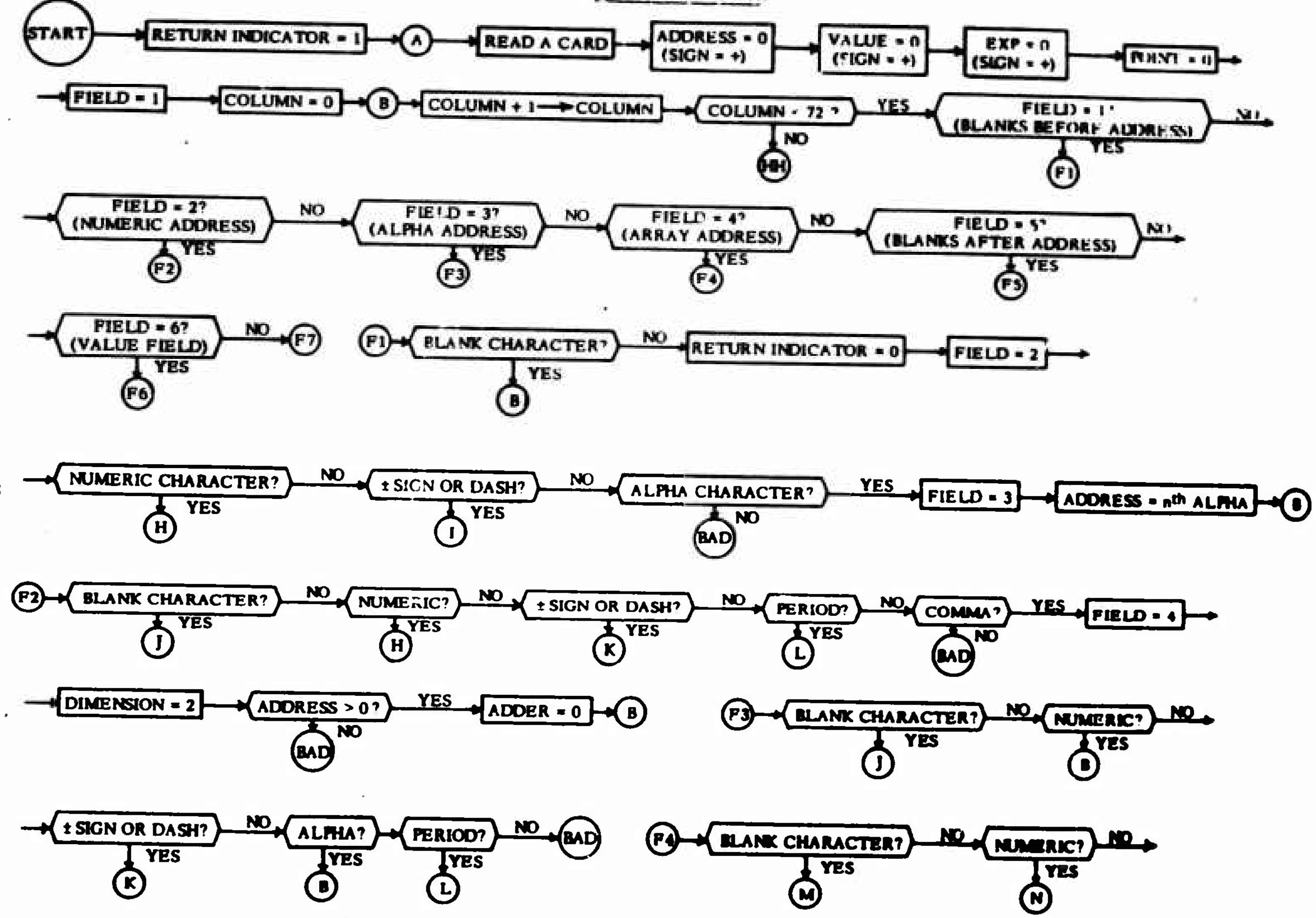
5. Method:

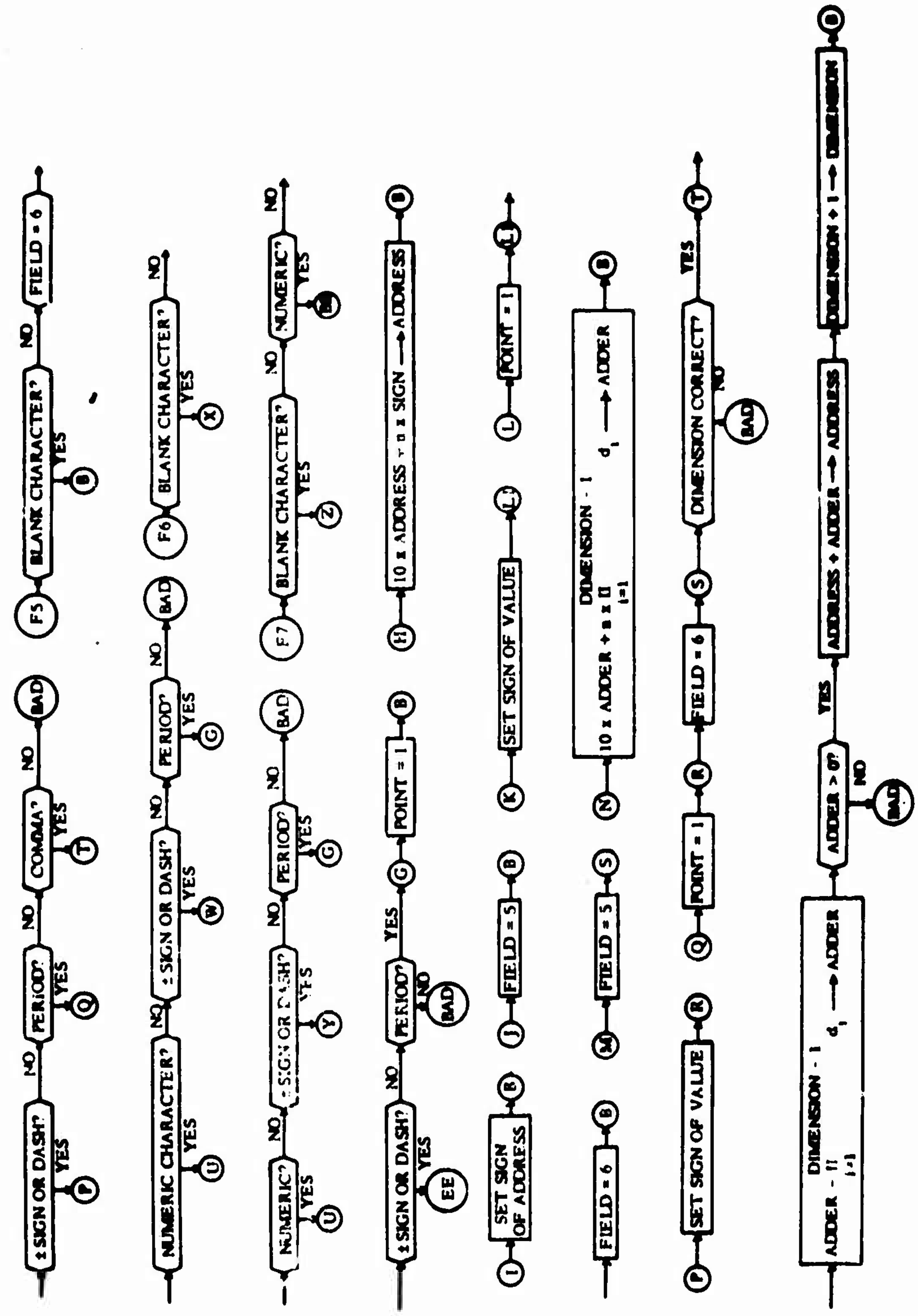
See the attached flow chart. DATA reads parameter values and loading addresses from cards. If sense switch 5 is up, it will read the values and addresses from tape (unit A2). It converts the values to floating point numbers, and stores them as elements of an array specified in the calling statement. The elements are specified by the addresses. If a card (or tape record) is read which contains non-permitted characters (see input card format above), DATA prints the statement "bad data card," followed by an image of the card itself.

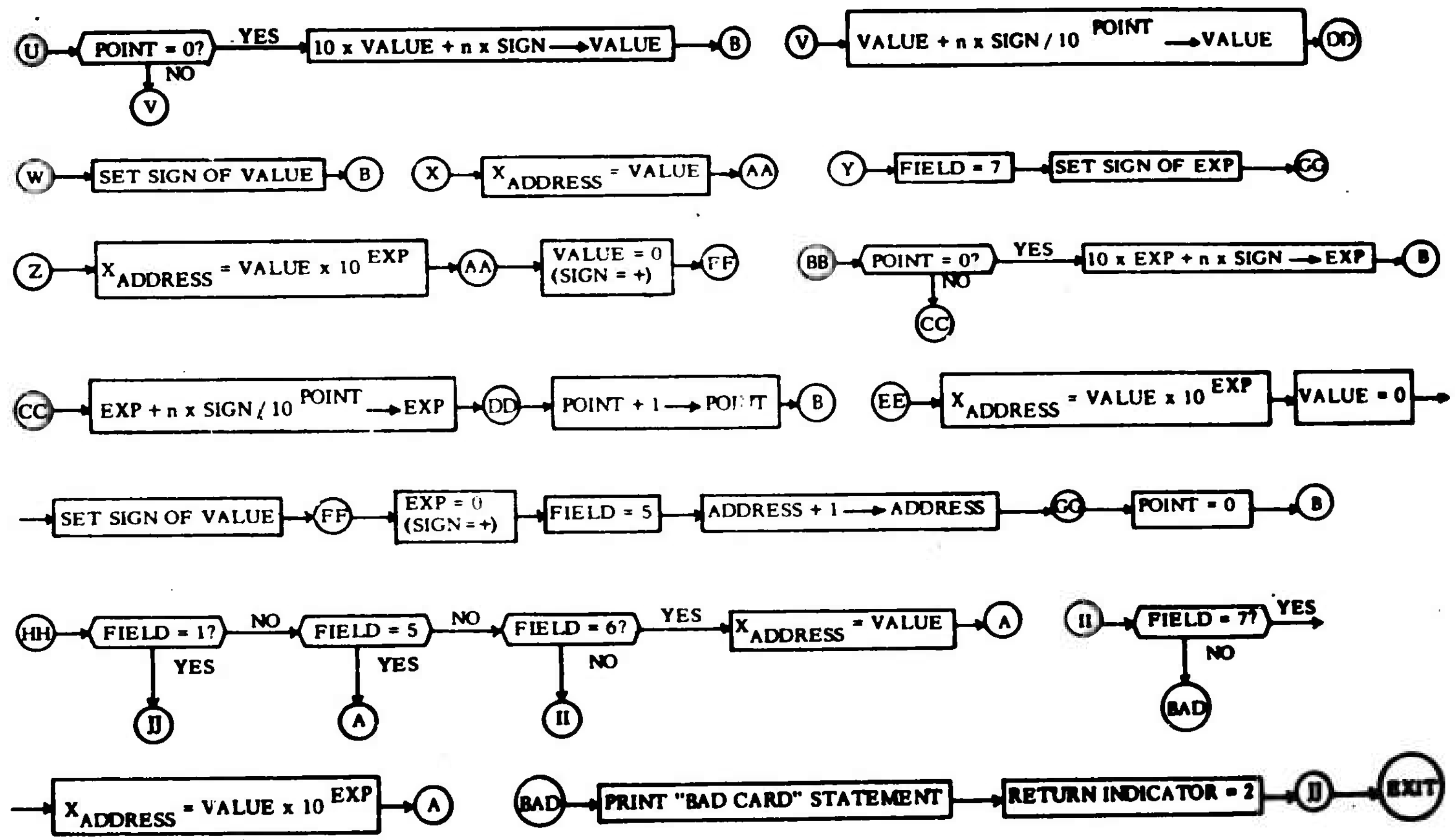
6. Coding Information:

See the symbolic listing included in this appendix. DATA is written in the 7090 FAP language. It must be used in conjunction with the BELL system. It requires 401 words storage space.

SUBROUTINE "DATA"




± SIGN OR DASH?
YES
P
NO
PERIOD?
YES
Q
NO
COMMA?
YES
T
NO
BAD
F5
BLANK CHARACTER?
YES
B
NO
FIELD = 6
NUMERIC CHARACTER?
YES
U
NO
± SIGN OR DASH?
YES
W
NO
PERIOD?
YES
G
NO
BAD
F6
BLANK CHARACTER?
YES
X
NO
NUMERIC?
YES
U
NO
± SIGN OR DASH?
YES
Y
NO
PERIOD?
YES
G
BAD
F7
BLANK CHARACTER?
YES
Z
NO
NUMERIC?
YES
NO
± SIGN OR DASH?
YES
EE
NO
PERIOD?
NO
BAD
YES
G
POINT = 1
B
H
10 x ADDRESS + n x SIGN → ADDRESS
B
I
SET SIGN OF ADDRESS
B
J
FIELD = 5
B
K
SET SIGN OF VALUE
L
POINT = 1
L
FIELD = 6
B
M
FIELD = 5
S
N
10 x ADDER + n x Π
DIMENSION - 1
d → ADDER
B
P
SET SIGN OF VALUE
R
Q
POINT = 1
R
FIELD = 6
S
DIMENSION CORRECT?
NO
BAD
YES
T
ADDER - Π
DIMENSION - 1
d → ADDER
ADDER > 0?
NO
BAD
YES
ADDRESS + ADDER → ADDRESS
DIMENSION + 1 → DIMENSION
D


U
POINT = 0?
YES
NO
V
10 x VALUE + n x SIGN → VALUE
B
V
VALUE + n x SIGN / 10 POINT → VALUE
DD
W
SET SIGN OF VALUE
B
X
X ADDRESS = VALUE
AA
Y
FIELD = 7
SET SIGN OF EXP
GG
Z
X ADDRESS = VALUE x 10 EXP
AA
AA
VALUE = 0
(SIGN = +)
FF
BB
POINT = 0?
YES
NO
CC
10 x EXP + n x SIGN → EXP
B
CC
EXP + n x SIGN / 10 POINT → EXP
DD
DD
POINT + 1 → POINT
B
EE
X ADDRESS = VALUE x 10 EXP
VALUE = 0
SET SIGN OF VALUE
FF
EXP = 0
(SIGN = +)
FIELD = 5
ADDRESS + 1 → ADDRESS
GG
POINT = 0
B
HH
FIELD = 1?
NO
YES
JJ
FIELD = 5
NO
YES
A
FIELD = 6?
YES
NO
II
X ADDRESS = VALUE
A
II
FIELD = 7?
YES
NO
BAD
X ADDRESS = VALUE x 10 EXP
A
BAD
PRINT "BAD CARD" STATEMENT
RETURN INDICATOR = 2
JJ
EXIT

SYMBOLIC LISTING

```
              FAP
              ENTRY DATA
      DATA    SXA X1,1
              SXA X2,2
              SXA X4,4
              CAL 1,4
              ADD CORE
              STO XLOC
              AXT 1,1
              SXA *+1,1
              CAL **,4
              ANA MASK
              TNZ *+2
              TXI *-4,1,1
              SXA EXIT,1
              TXI *+1,1,-1
              SXA *+1,1
              CLA **,4
              STA A1
              STA F1A
              STA I12
              AXT 1,1          RETURN INDICATOR = 1
      A1      SXD **,1
      A       TSX HHREAD,4     READ A CARD
              PZE CARD
              TRA EXIT
              TRA BAD
              STZ ADDRES       ADDRESS = 0
              STZ VALUE        VALUE = 0
              STZ EXP          EXP = 0
              STZ POINT        POINT = 0
              AXT 1,1          FIELD = 1
              SXA FIELD,1
              AXT 13,1
      A2      TNX HH,1,1       COLUMN GT 72
              AXT 42,2
              SXA COLUMN,2
      B       LXA COLUMN,2     COLUMN = COLUMN+1
              TNX A2,2,6
              SXA COLUMN,2
              LDQ CARD+12,1
              RQL 36,2
              PXD 0,0
              LGL 6
              STO CHARAC
              ORA FLOAT
              FAD FLOAT
              STO NUMB
              AXT 42,4
              CLA CHARAC
              CAS TABLE+42,4
              TRA *+2
              TRA *+3
              TIX *-3,4,1
```

```
        TRA  BAD
        LXA  FIELD,2
        TRA  F1+1,2
        TRA  F7          FIELD=7 (EXPONENT FIELD)
        TRA  F6          FIELD=6 (VALUE FIELD)
        TRA  F5          FIELD=5 (BLANKS AFTER ADDRESS)
        TRA  F4          FIELD=4 (ARRAY ADDRESS)
        TRA  F3          FIELD=3 (ALPHA ADDRESS)
        TRA  F2          FIELD=2 (NUMERIC ADDRESS)
F1      TXH  B,4,41      FIELD=1 (BLANKS BEFORE ADDRESS)
F1A     STZ  **          RETURN INDICATOR = 0
        AXT  2,2         FIELD = 2
        SXA  FIELD,2
        TXH  H,4,31      NUMERIC CHARACTER
        TXH  I,4,28      SIGN OR DASH
        TXL  BAD,4,2
        AXT  3,2         ALPHA CHARACTER, FIELD = 3
        SXA  FIELD,2
        TXI  *+1,4,-2    ADDRESS = NTH ALPHA
        SXA  ADDRES,4
        TRA  B
F2      TXH  J,4,41      BLANK CHARACTER
        TXH  H,4,31      NUMERIC CHARACTER
        TXH  K,4,28      SIGN OR DASH
        TXH  BAD,4,2
        TXH  L,4,1       PERIOD
        AXT  4,2         COMMA,FIELD = 4
        SXA  FIELD,2
        AXT  2,2
        SXA  DIMENS,2    DIMENSION = 2
        CLA  ADDRES      TEST ADDRESS
        TZE  BAD
        TMI  BAD
F2A     STZ  ADDER       ADDER=0
        TRA  B
F3      TXH  J,4,41      BLANK CHARACTER
        TXH  B,4,31      NUMERIC CHARACTER
        TXH  K,4,28      SIGN OR DASH
        TXH  B,4,2       ALPHA CHARACTER
        TXH  L,4,1       PERIOD
        TRA  BAD
F4      TXH  M,4,41      BLANK CHARACTER
        TXH  N,4,31      NUMERIC CHARACTER
        TXH  P,4,28      SIGN OR DASH
        TXH  BAD,4,2
        TXH  Q,4,1       PERIOD
        TRA  T           COMMA
F5      TXH  B,4,41      BLANK CHARACTER
        AXT  6,2         FIELD = 6
        SXA  FIELD,2
        TXH  U,4,31      NUMERIC CHARACTER
        TXH  W,4,28      SIGN OR DASH
        TXH  BAD,4,2
        TXH  G,4,1       PERIOD
```

```
        TRA  BAD
F6      TXH  X,4,41         BLANK CHARACTER
        TXH  U,4,31         NUMERIC CHARACTER
        TXH  Y,4,28         SIGN OR DASH
        TXH  BAD,4,2
        TXH  G,4,1          PERIOD
        TRA  BAD
F7      TXH  Z,4,41         BLANK CHARACTER
        TXH  BB,4,31        NUMERIC CHARACTER
        TXH  EE,4,28        SIGN OR DASH
        TXH  BAD,4,2
        TXL  BAD,4,1
G       AXT  1,2            PERIOD, POINT = 1
        SXA  POINT,2
        TRA  B
H       LDQ  ADDRES         ADDRESS = 10 X ADDRESS + N
        MPY  H10
        XCA
        ACL  CHARAC
        STO  ADDRES
        TRA  B
I       TXH  B,4,30         + SIGN
        CLA  ADDRES         SET SIGN OF ADDRESS
        SSM
        STO  ADDRES
        TRA  B
J       AXT  5,2            FIELD = 5
        SXA  FIELD,2
        TRA  B
K       TXH  L1,4,30        + SIGN
        CLA  VALUE          SET SIGN OF VALUE
        SSM
        STO  VALUE
        TRA  L1
L       AXT  1,2            POINT = 1
        SXA  POINT,2
L1      AXT  6,2            FIELD = 6
        SXA  FIELD,2
        TRA  B
M       AXT  5,2            FIELD = 5
        SXA  FIELD,2
        TRA  S
N       LDQ  ADDER          ADDER = 10 X ADDER + N X PROD
        MPY  H10
        STQ  ADDER
        TSX  T1,4
        MPY  CHARAC
        XCA
        ADD  ADDER
        STO  ADDER
        TRA  B
P       TXH  R,4,30         + SIGN
        CLA  VALUE          SET SIGN OF VALUE
        SSM
```

```
        STO VALUE
        TRA R
Q       AXT 1,2          POINT = 1
        SXA POINT,2
R       AXT 6,2          FIELD = 6
        SXA FIELD,2
S       LXA EXIT,2       CHECK DIMENSION
        TXI *+1,2,-3
        PXA 0,2
        SUB DIMENS
        TNZ BAD
T       TSX T1,4         ADDER=ADDER-PROD
        CLA ADDER
        SUB PROD
        STO ADDER
        TZE BAD          CHECK ADDER
        TMI BAD
        ADD ADDRES
        STO ADDRES
        CLA DIMENS
        ADD H1
        STO DIMENS
        TRA F2A
T1      SXA T4,4         PROD = PRODUCT OF DIMENSIONS
        CLA H1
        STO PROD
        STA T3
        LXA DIMENS,2
        TXI *+1,2,-1
        LXA X4,4
T2      CAL T3
        ADD H1
        STA T3
T3      CLA **,4
        STA *+1
        LDQ **
        RQL 18
        MPY PROD
        STQ PROD
        TIX T2,2,1
T4      AXT **,4
        TRA 1,4
U       CLA POINT        TEST POINT
        TNZ V
        LDQ VALUE        VALUE = 10 X VALUE + N
        FMP DEC10
        SSP
        FAD NUMB
        LDQ VALUE
        LLS 0
        STO VALUE
        TRA B
V       LXA POINT,4      VALUE = VALUE + N/(10**POINT)
        CLA NUMB
```

```
         FDP DEC10
         XCA
         TIX *-2,4,1
         LDQ VALUE
         LLS 0
         FAD VALUE
         STO VALUE
         TRA DD
W        TXH B,4,30        + SIGN
         CLA VALUE         SET SIGN OF VALUE
         SSM
         STO VALUE
         TRA B
X        CLA XLOC          X(ADDRESS) = VALUE
         SUB ADDRES
         STA *+2
         CLA VALUE
         STO **
         TRA AA
Y        AXT 7,2
         SXA FIELD,2       FIELD = 2
         TXH GG,4,30       + SIGN
         CLA EXP           SET SIGN OF EXP
         SSM
         STO EXP
         TRA GG
Z        CLA XLOC          X(ADDRESS) = VALUE X 10**EXP
         SUB ADDRES
         STA Z1
         CLA DEC10
         LDQ EXP
         CALL EXP(3
         XCA
         FMP VALUE
Z1       STO **
AA       STZ VALUE         VALUE = 0
         TRA FF
BB       CLA POINT         TEST POINT
         TNZ CC
         LDQ EXP           EXP = 10 X EXP + N
         FMP DEC10
         SSP
         FAD NUMB
         LDQ EXP
         LLS 0
         STO EXP
         TRA B
CC       LXA POINT,4       EXP = EXP + N/(10**POINT)
         CLA NUMB
         FDP DEC10
         XCA
         TIX *-2,4,1
         LDQ EXP
         LLS 0
```

```
         FAD EXP
         STO EXP
DD       CLA POINT          POINT = POINT + 1
         ADD H1
         STO POINT
         TRA B
EE       CLA XLOC           X(ADDRESS) = VALUE X 10**EXP
         SUB ADDRES
         STA EE1
         CLA DEC10
         LDQ EXP
         CALL EXP(3
         XCA
         FMP VALUE
EE1      STO **
         PXD 0,0            VALUE = 0
         TXH *+2,4,30       + SIGN
         SSM                SET SIGN OF VALUE
         STO VALUE
FF       STZ EXP            EXP = 0
         AXT 5,2            FIELD = 5
         SXA FIELD,2
         CAL ADDRES         ADDRESS = ADDRESS + 1
         ADD H1
         SLW ADDRES
GG       STZ POINT          POINT = 0
         TRA B
HH       LXA FIELD,1
         TXL JJ,1,1         FIELD=1, EXIT
         TXL BAD,1,4
         TXL A,1,5          FIELD=5, READ ANOTHER CARD
         TXH II,1,6
         CLA XLOC           FIELD=6, X(ADDRESS) = VALUE
         SUB ADDRES
         STA *+2
         CLA VALUE
         STO **
         TRA A
II       TXH BAD,1,7
         CLA XLOC           FIELD=7,
         SUB ADDRES         X(ADDRESS) = VALUE X 10**EXP
         STA III
         CLA DEC10
         LDQ EXP
         CALL EXP(3
         XCA
         FMP VALUE
III      STO **
         TRA A
BAD      TSX HPRINT,4
         PZE PRINT,0,15
         AXT 2,1
II2      SXD **,1
X1       AXT **,1
```

```
X2      AXT **,2
X4      AXT **,4
EXIT    TRA **,4
MASK    OCT 777777700000
PRINT   BCD 3 BAD DATA CARD...
CARD    BSS 12
ADDRES  HTR **
VALUE   HTR **
EXP     HTR **
POINT   HTR **
FIELD   HTR **
COLUMN  HTR **
TABLE   OCT 60              BLANK
        OCT 0               0
        OCT 1               1
        OCT 2               2
        OCT 3               3
        OCT 4               4
        OCT 5               5
        OCT 6               6
        OCT 7               7
        OCT 10              8
        OCT 11              9
        OCT 20              + SIGN
        OCT 40              - SIGN
        OCT 14              DASH
        OCT 71              Z
        OCT 70              Y
        OCT 67              X
        OCT 66              W
        OCT 65              V
        OCT 64              U
        OCT 63              T
        OCT 62              S
        OCT 51              R
        OCT 50              Q
        OCT 47              P
        OCT 46              O
        OCT 45              N
        OCT 44              M
        OCT 43              L
        OCT 42              K
        OCT 41              J
        OCT 31              I
        OCT 30              H
        OCT 27              G
        OCT 26              F
        OCT 25              E
        OCT 24              D
        OCT 23              C
        OCT 22              B
        OCT 21              A
        OCT 33              PERIOD
        OCT 73              COMMA
```

```
CHARAC HTR **
DIMENS HTR **
 ADDER HTR **
 H10    HTR 10
 DEC10 DEC 10.0
 H1     HTR 1
 PROD  HTR **
 AMASK OCT 77777
 FLOAT OCT 233000000000
 NUMB  HTR **
 XLOC  HTR **
 CORE  OCT 100001
 JJ     SYN X1
        END
```

OEG COMPUTER DATA SUBMITTAL FORM

Submitted by: ______________________________ **Date:** ______________

Program No. ____________ **Est. Time** ____________ **Classification** __________

Special Instructions: ____

Address	Value	Address	Value	Address	Value	Address	Value

NOTES:

1. A value of zero must be entered as 0, not left blank.
2. Decimal pts. may be omitted if understood to follow the rightmost digit.
3. The value 3 X 10^{-5} may be entered as .00003 or 3-5, not as 3 X 10^{-5}.
4. The factor portion of a value may not contain more than 8 digits.
5. The exponent portion of a value must lie within the range ±39.
6. Exponents may be omitted if zero. If not, they must be signed.
7. Blank cards should be indicated by: ———————b——————— .

None

Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title body of abstract and indexing annotation must be entered when the overall report is classified)

1 ORIGINATING ACTIVITY (Corporate author) Operations Evaluation Group, Center for Naval Analyses of the Franklin Institute	2a REPORT SECURITY CLASSIFICATION None 2b GROUP None
3 REPORT TITLE Complex Target Coverage OEG Computer Program 13-63P	
4. DESCRIPTIVE NOTES (Type of report and inclusive dates) Revised version (26 March 1965) of earlier publication.	
5 AUTHOR(S) (Last name, first name, initial) Smith, R.L., Westlund, G.A., DePoy, P.E., Ridings, R.V., and Denenberg, S.A.	
6. REPORT DATE 26 March 1965	7a TOTAL NO OF PAGES 50 7b NO OF REFS 2
8a. CONTRACT OR GRANT NO. NONR 3732 (00) b. PROJECT NO ----- c. ----- d. -----	9a ORIGINATOR'S REPORT NUMBER(S) OEG Research Contribution No. 68 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) -------
10. AVAILABILITY/LIMITATION NOTICES Qualified requestors may obtain from DDC. Not cleared for public release.	
11. SUPPLEMENTARY NOTES Supercedes OEG Interim Research Memorandum No. 45	12. SPONSORING MILITARY ACTIVITY Office of Naval Research Navy Department Washington 25, D.C.

13. ABSTRACT

This research contribution presents a usage manual for an IBM 7090 computer program. The program employs a Monte Carlo simulation to determine the probability of destroying individual point targets within a target complex with one or more groups of weapons. It is assumed that the groups are delivered with a bivariate normal aiming error and that the individual weapons are distributed with an independent bivariate normal ballistic dispersion. The program is designed for conditional damage data for fragmentation generated by an IBM 7090 program furnished by the U.S. Naval Ordnance Test Station (NOTS), China Lake. A flow chart, a listing of the FORTRAN program and a sample problem are included.

DD FORM 1 JAN 64 1473

None

Security Classification

None
Security Classification

14 KEY WORDS	LINK A ROLE	LINK A WT	LINK B ROLE	LINK B WT	LINK C ROLE	LINK C WT
Probability of target destruction						
Destruction of point targets in target complex						
Monte Carlo simulation						
IBM 7090 computer program						
FORTRAN						

INSTRUCTIONS

1. **ORIGINATING ACTIVITY:** Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization (*corporate author*) issuing the report.

2a. **REPORT SECURITY CLASSIFICATION:** Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2b. **GROUP:** Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. **REPORT TITLE:** Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. **DESCRIPTIVE NOTES:** If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. **AUTHOR(S):** Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. **REPORT DATE:** Enter the date of the report as day, month, year; or month, year. If more than one date appears on the report, use date of publication.

7a. **TOTAL NUMBER OF PAGES:** The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. **NUMBER OF REFERENCES:** Enter the total number of references cited in the report.

8a. **CONTRACT OR GRANT NUMBER:** If appropriate, enter the applicable number of the contract or grant under which the report was written.

8b, 8c, & 8d. **PROJECT NUMBER:** Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9a. **ORIGINATOR'S REPORT NUMBER(S):** Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9b. **OTHER REPORT NUMBER(S):** If the report has been assigned any other report numbers (*either by the originator or by the sponsor*), also enter this number(s).

10. **AVAILABILITY/LIMITATION NOTICES:** Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through ________________."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through ________________."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through ________________."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. **SUPPLEMENTARY NOTES:** Use for additional explanatory notes.

12. **SPONSORING MILITARY ACTIVITY:** Enter the name of the departmental project office or laboratory sponsoring (*paying for*) the research and development. Include address.

13. **ABSTRACT:** Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS)*, *(S)*, *(C)*, or *(U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. **KEY WORDS:** Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, roles, and weights is optional.

DD FORM 1 JAN 64 1473 (BACK)

None
Security Classification